VOLUMETRIC REPRESENTATION FOR SPARSE MULTI-VIEWS

N. Anantrasirichai, C. Nishan Canagarajah , David W. Redmill and David R. Bull

Department of Electrical & Electronic Engineering,
University of Bristol, Bristol, BS8 1UB, UK

ABSTRACT

In this paper, we propose a novel volumetric representation for a sparse set of calibrated multi-view images of a non-Lambertian scene. The depth map of each reference view is registered into a volume and a simple algorithm to shape the volume is introduced. Particular colours are defined for each voxel to render a smooth and realistic image. Synthesized results demonstrate the good performance with the proposed scheme, both for parallel-camera and non-parallel-camera geometries.

Index Terms— Three-dimensional vision, Image synthesis

1. INTRODUCTION

Nowadays three-dimensional (3D) visual environments have become one of the emerging problems in communication and computer vision. Multiple cameras are used to obtain object or scene information from various points of view. This amount of data has limited many commercial applications, due to a manifold increase in bandwidth over existing monoscopic image system. Thus, the efficient compression algorithms are vital to reduce the size of data without sacrificing the perceived image quality. As the sequences are captured from the same scene, a 3D geometric representation is a good solution to remove redundant data. One such representation is a *volumetric representation*. Obviously, if a greater number of cameras are used, this method will be more effective in terms of both compression ratio and modelling accuracy, i.e. it is also very efficient for virtual view synthesis. The compression of this single reconstructed volume can be efficiently achieved by exploiting 3D coding schemes [1][2][3].

Algorithms for 3D reconstruction from multi-views are continually proposed and it is possible to classify them into two categories. The first is *shape-from-silhouette* algorithms which are based on volume intersection [4]. The binary data taken from an object's boundary are registered into a volume. The algorithm needs a very narrow-baseline geometry and therefore requires a high number of views. The algorithm has been developed by exploiting grey scale. With each voxel it must be determined whether it is transparent or whether it is one of the volume by following photometric constraints. Then, the object's geometry is used to refine the surface [5]. Other methods use previously estimated depth maps to construct a volume. Each depth map is warped to the geometry of the volume and the opaque or the transparent voxels are easily marked. This makes accurate depth maps necessary [6][7]. Several algorithms have been proposed to manage the error and noise in the depth maps. In [8], the uncertainty of depth values is handled with a boundary condition; however, it was proposed for a static scene captured by moving a single camera. It is not straightforward to apply this to a multi-view video system. The statistical method introduced in [9] needs a lot of input data to rectify an object's surface using a probability distribution. In [10], a Bayesian MAP problem is formulated to lead to the energy minimization that could support a sparse view environment but it was proposed under the assumption of Lambertian reflection.

Most 3D reconstruction techniques have been developed for specific scenarios. They require a great number and variety of views and/or they are initiated with colour consistency constraints. In a real-life 3D video system using a single camera is not possible and using a great many cameras is too expensive. In this paper we consider a sparse multi-view system where all the cameras may be not identical, and may not converge to a single common point in the scene. We construct the volume by registering the estimated depth maps. Not only is this simple and fast, but it also provides a degree of freedom to select the appropriate depth estimation approach. Each view is searched for the depth along its own geometry. Therefore it is possible to compensate for non-Lambertian reflectance by exploiting the individual minimum cost to obtain the depth value instead of defining a global threshold. Moreover, depth registration methods can support a good representation for the natural scenery.

We also propose a novel algorithm to refine the shape of the volume. The constructed volume should accurately represent the real scene or object. Its incorrect shape could adversely affect the virtual view synthesis. However, sharp corners always consume a quantity of coding data. We therefore adjust the surface by considering the neighbouring voxels. Finally, colours are defined for each voxel. To ensure a smooth *look around* capability, a good colour selection method should compensate for several sources of

noise, e.g. camera calibration and imaging, imperfect light balance and volume quantization. In this paper, the effect of a smooth volume is investigated by checking the subjective quality of synthesized views.

The rest of paper is organized as follows: Section 2 briefly explains the fundamentals of 3D geometry and some problems in building up a volume. The proposed scheme is described in Section 3. The experimental results are presented in Section 4. Conclusions and future work follow in Section 5.

2. 3D GEOMETRY AND PROBLEMS

In this paper, we consider the reconstruction of a 3D object or a natural scene from a set of 2D images taken from several point of views. We assume that the camera parameters are known or obtainable by a camera calibration algorithm. The depth map for each view has been estimated, with any uncertainty and noise still partly present. The association between a 3D volume and a set of 2D-image geometries can be stated as follows. A pixel $\mathbf{x}_m = [i_m, j_m]^{\mathrm{T}}$ in image view m, which has depth w_{ij}^m and the geometry parameters $\mathbf{P}_m$, is mapped to a voxel position $\tilde{\mathbf{x}} = [\tilde{i}, \tilde{j}, \tilde{k}]^{\mathrm{T}}$ with the following invertible relation: $[\tilde{i}\tilde{k},\ \tilde{j}\tilde{k},\ \tilde{k}]^{\mathrm{T}} = \tilde{\mathbf{P}}\,\mathbf{P}_m^{-1}\,[i_m w_{ij}^m,\ j_m w_{ij}^m,\ w_{ij}^m]^{\mathrm{T}}$, where $\tilde{\mathbf{P}}$ is the geometry parameters of the volume. If the calibration is perfect and the medium meets Lambertian reflection, using the above equation to construct a volume is not difficult. However, it does not provide the perfect matching in real scenarios where some problems, due to imperfect light balance, non-Lambertian surfaces and different resolutions, cannot be avoided.

With photometric constraints, the intensity of $\mathbf{x}_m$ must be the same as that of $\mathbf{x}_n$ if they are projected from the same voxel $\tilde{\mathbf{x}}$. Therefore, a voxel that does not satisfy the photometric constraints should be transparent or be removed from the volume. However, this is only true for Lambertian fronto-parallel surfaces in which the corresponding pixels on all visible views have the same intensity. For a non-Lambertian scene, the colour constraints are included, since the colour components still have the same tendency in reflecting medium. In addition, a tolerance must be allowed for intensity matching.

The cameras are placed at various angles and the object of interest might not be at the centre of the scene. This causes different sizes of the object in each captured image. There are two problems caused by these unequal resolutions which lead to smaller or larger sizes when compared to the size of the constructed volume. If the 2D object in the arbitrary view is smaller, the adjacent pixels are possibly mapped to the non-adjacent voxels. Hence, an interpolation is required to connect it as a continuous surface. On the other hand if the 2D object is larger, two or more pixels might associate with a single voxel. This case can cause a major problem as colours belonging to such pixels are totally different. Moreover, if only one colour is defined to a voxel, the projected pixel in one image might be completely different from the original one. This makes colour selection process difficult.

An important aspect of this work is that it should produce a smooth and realistic image. Therefore, the colour defined under the visibility constraints should satisfy the smoothness constraint. That means the artificial colours sometimes give better subjective quality and better viewing experience than the true colours.

3. PROPOSED SCHEME

This section describes the proposed volumetric representation scheme. A volume is reconstructed from a set of depth maps. After refining, colours are post-processed.

3.1. Volume Initialization

Firstly, depth maps are registered into a volume. Each registered voxel has a value equal to the number of views that are mapped to this voxel. Then, the unmapped voxels have a value of 0. The voxel value $\delta_{\tilde{i}\tilde{j}\tilde{k}}$ can be expressed as follows;

$$\delta_{\tilde{i}\tilde{j}\tilde{k}} = \begin{cases} n, & \textit{if registering depth,} \\ 0, & \textit{otherwise} \end{cases} \tag{1}$$

where the number of views that are mapped to a voxel $\tilde{\mathbf{x}}$ is n for $n \in \{1, 2, \ldots, N\}$, with total N reference views.

Then some isolated voxels which do not have any neighbours are detected. These have a high possibility of being errors; therefore they should be removed from the volume. Next, the volume is projected back to the reference view geometry to build up new depth maps that now include the depth information from other views that some errors were removed in the previous step. If there is more than one depth value for a pixel, the front cluster of depths is considered. Ideally, there is one depth value projected from the front surface, but spurious noise makes it dispersed. The proper depth value $\tilde{d}$ of a particular pixel is

$$\tilde{d} = \begin{cases} d_n, & \textit{if only one depth has } \delta_n = \max(\delta), \\ \left(\sum_{k=1}^{K} \delta_k d_k\right) / \left(\sum_{k=1}^{K} \delta_k\right), & \textit{otherwise} \end{cases} \tag{2}$$

where K is the number of depth values in the front cluster. The depth value projected from the voxel that contains the maximum δ is selected. If the maximum δ appears in two or more depth values, the fit depth value is calculated from the average of all depth values of the front cluster with weighted coefficient δ.

Subsequently, the volume is reconstructed again. In this step, some voxels are interpolated in order to connect the surface of the volume among the near voxels. This is

necessary because the available reference views have different resolution as discussed in Section 2.

3.2. Volume Refinement

This step is to refine the volume by iterating several times. We reconstruct all views from the volume with colours and compare it with the original reference images. The colours are temporarily defined by taking the median of colour values from all the visible views. Note that the median is less sensitive to extreme values than the mean and this makes it a better measure for highly skewed distributions.

The iterative process runs from the first reference view, probably the furthest left view, to the last reference view, probably the furthest right view. For each view, the mean square errors (MSE) between the reconstructed view and the original view are calculated. If the error is more than a threshold, the old depth is removed and it also causes the associated voxel to be removed. Then the new depth, which is derived from the neighbouring pixels (a smooth surface assumption) is investigated. If this new depth value meets colour constraints for all visible views, the new associated voxel is generated. It is noticeable that the imperfect camera calibration and unequal resolutions of each of the reference views possibly make a mapped pixel be displaced from its exact position. A window-based computation, e.g. 3x3 pixels, therefore gives a better result for error calculation than exploiting only one pixel.

If some voxels are removed or newly generated, we iteratively reconstruct the images repeatedly. This is because the change of the position of a voxel produces a new depth, which might change an invisible point to a visible point and vice versa.

3.3. Colour Selection

From previous step, a fine volume is constructed. This section illustrates the post-processing colour selection for each voxel. Due to non-Lambertian reflectance, the temporary colours defined in the iterative volume refinement are probably not suitable. Smooth colours and brightness are needed, hence we suggest using the weighted average of the colours from one or two of the nearest visible views for defining voxel colour $\widetilde{C}$.

First of all, the line L_c of volume's centre is approximately marked. Then the line L_v along the volume surface that is the intersection of the surface and the line created from L_c to the camera centre $\mathbf{c}_v$ of view v are drawn as illustrated in Fig. 1.

By scanning along L_c, the voxels of each layer obtain the colours as Equation 3 with a *condition* that a voxel $\widetilde{x}$ is visible on at least two views and it is located between L_1 and L_2. The first two nearest visible views are view1 and view2

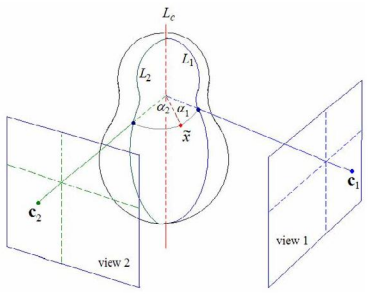


Fig. 1 an example of the marked line L_v (v =1,2) that is used to calculate the distance for colour defining.

respectively.

$$\widetilde{C} = \begin{cases} (\alpha_1 C_2 + \alpha_2 C_1)/(\alpha_1 + \alpha_2) & \textit{if the condition is true,} \\ C_1 & \textit{otherwise} \end{cases} \quad (3)$$

where C_1 and C_2 represent the intensity component (Y) or the colour components (U,V) of view 1 and view 2. α_1 and α_2 are the angles between the voxel $\widetilde{x}$ and the point in the same layer as such voxel on L_1 and L_2 respectively.

The average method might cause object blur, if C_1 and C_2 are too different. Therefore, the defined colours are set into a *blur* group, if the difference of C_1 and C_2 is more than a particular threshold, or a *clear* group, if the difference of C_1 and C_2 is less than or equal the threshold. To preserve sharpness, colour $\widetilde{C}_{old}$ of the voxel in the *blur* group is replaced by colour of neighbouring voxel that has the closest value to $\widetilde{C}_{old}$ and has to be in the *clear* group.

4. EXPERIMENTAL RESULTS

The proposed scheme was tested with a non-parallel-geometry sequence, *Leo*[1], composed of five cameras placed in a horizontal direction. The original images of the furthest left view, the left view and the middle view are shown in Fig. 2. The object, a toy standing on two video tapes, is produced with different resolutions in each view.

Firstly, we extracted the foreground from the background for each image by checking the depth values and the gradient of depth values around the object boundaries. The foreground depth maps are then applied to the volume initialization process and the background was operated separately.

The initial volume is illustrated in Fig. 4 (a) by registering the estimated depth maps (the depth maps of images Fig. 2 are illustrated in Fig. 3). Many of the separated voxels are removed. Then, the volume was iteratively refined until no more voxel modification is performed. The result is shown in Fig. 4 (b).

[1] The multi-view *Leo* sequence was captured at University of Bristol

(a) (b) (c)

Fig. 2 The original *Leo* images; (a) the furthest left view (b) the left view (c) the middle view.

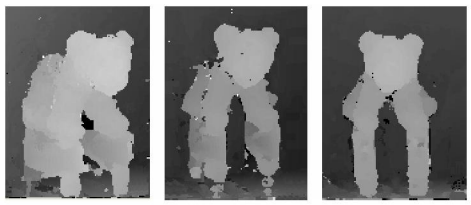
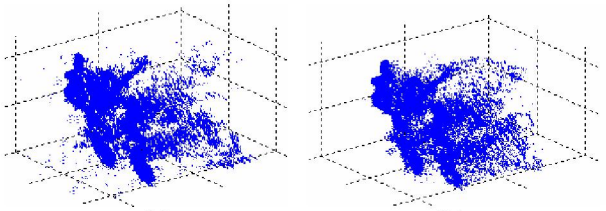


Fig. 3 The estimated depth maps of images in Fig. 2.



(a) (b)

Fig. 4 The voxel clouds after (a) depth maps were registered, (b) iterative volume refinement.

Finally, the colours are defined for the volume by exploiting the proposed scheme as explained in Section 3.3. The synthesized views are shown in Fig. 5. The pictures clearly show that the proposed scheme improves the represented volume and gives better results in synthesis. The object in Fig. 5 (c) looks sharper at boundaries and smoother in the homogenous areas.

(a) (b) (c)

Fig. 5 The results of view synthesis. (a) no enhancement applied. (b) the proposed volume reconstruction with median colour selection (c) the proposed volume reconstruction with the proposed colour selection.

The proposed scheme was also tested with a parallel-geometry sequence, *Santa*[1]. For this sequence, we have not extracted the foreground from the background. The complete depth maps are warped to a volume. This sequence shows that the proposed scheme works well, even though the background is included in the volume.

[1] from University of Tsukuba

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a novel volumetric representation by exploiting the information in estimated depth maps. It is useful for real environment, i.e. non-Lambertian scenes, and also practical if a small number of reference views are available. A volume, which can be an object or real scenery, is constructed by registering depth maps. Then this volume is refined by removing outlying voxels. By iteration, the voxels in the volume are further removed, if they produce high error compared to the original images. The new depths and associated voxels are generated. Finally particular colours are defined for each voxel to make the viewer perceive a smooth and realistic image. The results show that the proposed approach can lead to significant improvements in both parallel-camera and non-parallel-camera geometries. This proposed scheme is practical for multi-view video, therefore video sequences will be tested in the future.

6. REFERENCES

[1] J. Zhang, C.B. Owen, "Octree-based animated geometry compression," *IEEE Proc. DCC'04*, pp. 508-517, 2004.

[2] F.F Rodler, F.F, "Wavelet based 3D compression with fast random access for very large volume data," *IEEE Proc. 7th Pacific Conf. on Computer Graphics and Applications*, pp. 108-117, 1999.

[3] B. Felts, B. Pesquet-Popescu, "Efficient context modeling in scalable 3D wavelet-based video compression," *IEEE Proc. ICIP'00*, vol. 1, PP. 1004-1007, 2000.

[4] E. Boyer, "Object models from contour sequences," *Proc. ECCV'96*, pp. 109-118. 1996.

[5] P. Eisert, E. Steinbach and B. Girod, "Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.10 , pp.261-277, 2000.

[6] P. Beardsley, P. Torr, and A. Zisserman, "3D model acquisition from extended image sequences," *Proc. ECCV'96*, UK, pp. 683-695, 1996.

[7] R. Koch, M. Pollefeys and L. Van Gool, "Realistic 3D Scene Modeling from Uncalibrated Image Sequences," *IEEE Proc. ICIP'99*, vol.2 , pp. 500-504, 1999.

[8] F.E. Ernst, C.W.A.M. van Overveld, P. Wilinski, " Efficient generation of 3-D models out of depth maps," *Proc. VMV'01*, Germany, pp. 203-210, 2001.

[9] R. Koch, M. Pollefeys and L. Van Gool, "Robust Calibration and 3D Geometric Modeling from Large Collections of Uncalibrated Images," *Proc. DAGM'99*, pp. 413-420, 1999.

[10] P. Gargallo and P. Sturm, "Bayesian 3D Modeling from Images using Multiple Depth Maps," *Proc. CVPR'05*, vol.2, pp. 885-891, 2005.